

How to use the stat command on Linux

The Linux `stat` command shows you more details than the `ls` command. The following article will show you how to use the `stat` command in Linux. Like `ls`, the `stat` command has many options. This makes it a great candidate for the use of alias.

The Linux `stat` command shows you more details than the `ls` command. The following article will show you how to use the `stat` command in Linux.

Like `ls`, the `stat` command has many options. This makes it a great candidate for the use of alias. This makes using it much more convenient and you don't need to remember a complicated set of command line options.

Learn about the stat command on Linux

1. What does the `stat` command tell me?
2. Use `stat` with multiple files
3. Symbolic links
4. Report Terse
5. Custom output format

What does the stat command tell me?

Take a look at the following `stat` command:

```
stat ana.h
```

```
dave@howtogeek:~/ana$ stat ana.h
  File: ana.h
  Size: 802          Blocks: 8          IO Block: 4096   regular fi
le
Device: 801h/2049d Inode: 1442997    Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   dave)   Gid: ( 1000/   dav
e)
Access: 2019-12-13 11:40:13.927001000 -0500
Modify: 2015-12-13 07:20:41.490221000 -0500
Change: 2019-12-13 11:38:02.567001000 -0500
 Birth: -
dave@howtogeek:~/ana$
```

The information we have is:

1. **File:** The name of the file.

2. **Size:** The file size is in bytes.
3. **Blocks:** The number of file system blocks required, to store on the hard drive.
4. **IO Block:** The size of the file system block.
5. **File type:** The type of object that the metadata describes.
6. **Device:** The device number is in hexadecimal and decimal.
7. **Inode:** Number of inode. That is the ID number of this inode.
8. **Links:** This figure shows how many hard links point to this file.
9. **Access:** File permissions are displayed in traditional read, write, and execute formats.
10. **Uid:** The user ID and account name of the owner.
11. **Gid:** The group ID and the account name of the owner.
12. **Access:** Mark the access time.
13. **Modify:** Mark the modified time. This is the time when the file content was last modified.
14. **Change:** Mark the time of change. This is the time when the file's properties or contents were last modified.
15. **Birth:** Set to show the original creation date of the file, but this is not done in Linux.

Use stat with multiple files

To have `stat` reporting on several files at once, pass the file name to the `stat` on the command line:

```
stat ana.h ana.o
```

To use the `stat` on a set of files, use the appropriate pattern. Question mark '?' represent any single character and the asterisk '*' represent any string of characters.

```
dave@howtogeek:~/ana$ stat ana.h ana.o
  File: ana.h
  Size: 802          Blocks: 8          IO Block: 4096   regular fi
le
Device: 801h/2049d  Inode: 1442997     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   dave)  Gid: ( 1000/   dav
e)
Access: 2019-12-13 11:40:13.927001000 -0500
Modify: 2015-12-13 07:20:41.490221000 -0500
Change: 2019-12-13 11:38:02.567001000 -0500
  Birth: -
  File: ana.o
  Size: 18912       Blocks: 40         IO Block: 4096   regular fi
le
Device: 801h/2049d  Inode: 1442998     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/   dave)  Gid: ( 1000/   dav
e)
Access: 2019-12-13 11:21:21.495001000 -0500
Modify: 2019-04-19 08:10:47.154301000 -0400
Change: 2019-12-13 11:21:55.691001000 -0500
  Birth: -
```

Use stat to report on the file system

The `stat` command can report the status of the file systems, as well as the status of the files. The `-f` (filesystem) option requires the `stat` to report on the file system on which the file resides. Note that you can also move a directory like '/' to the `stat` instead of the file name.

```
stat -f ana.c
```

The `stat` command tells us:

1. **File:** The name of the file.
2. **ID:** The file system ID in hexadecimal notation.
3. **Namelen:** The maximum length allowed for file names.
4. **Type:** File system type.
5. **Block size:** The amount of data required to read for optimal data transfer rate.
6. **Fundamental block size :** The size of each file system block.
7. **Blocks:**
 1. Total: The total number of all blocks in the file system.
 2. Free: The number of free blocks in the file system.
 3. Available: Number of free blocks available to normal users (not root).
8. **Inodes:**
 1. Total: The total number of inodes in the file system.
 2. Free: The number of empty inodes in the file system.

Symbolic links

If you use the `stat` on a file that is actually a symbolic link, it will report on that link. If you want the `stat` to report on the file the link points to, use the `-L` (dereference) option. The file `code.c` is a symbolic link to `ana.c`. Consider it when there is no `-L` option:

```
stat code.c
```

The file name shows **code.c** pointing to (->) **ana.c**. File size is only 11 bytes. There are no blocks available for storing this link. The file type is listed as a symbolic link.

Run the command again and add the `-L` option :

```
stat -L code.c
```

The detailed display results for the file are indicated by the symbolic link. But note that the file name is still provided as `code.c`. This is the name of the link, not the target file.

Report Terse

The `-t` (terse) option causes the `stat` to provide a concise summary:

```
stat -t ana.c
```

No clues were given. To understand what it means, until you memorize the field sequence, you need to cross-reference this output to the full output.

Custom output format

A better way to get a different set of data from the `stat` is to use a custom format. There is a long list of tokens called a format string. Each of these represents a data element. Select the ones you want to have in the output

and create a format string. When you call the `stat` and pass the format string to it, the output will only include the data components you request.

There are two options for accepting a format string **--format** and **--printf**. The difference between them is **--printf** interprets the escape sequences of type C and it does not automatically add a newline character to the output.

Let's create a format string and convert it to `stat`. The format order to use is **% n** for the file name, **% s** for the file size, and **% F** for the file type. Add an escape string to the end of the string to ensure each file is processed on a new line:

```
"File %n is %s bytes, and is a %Fn"
```

The example will convert this to a `stat` using the **--printf** option .

```
stat --printf="File %n is %s bytes, and is a %Fn" code.c ana/ana.?
```

Reports for each file are listed on a new line.

Custom formats allow you to access more data elements than are included in the standard `stat` output.

As you can see, there is plenty of scope for extracting specific data elements that interest you.

You finished reading the article "**How to use the stat command on Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.